AD-A050 224 WASHINGTON UNIV SEATTLE DEPT OF CHEMISTRY F/G 9/2
A LOW-COST, GENERAL PURPOSE DATA ACQUISITION AND CONTROL SYSTEM--ETC(U)
FEB 78 J D DANIELSON, S D BROWN, C J APPELLOF N00014-75-C-0536
UNCLASSIFIED TR-11 NL

1 OF 1
AD
A050 224

END
DATE
FILMED
3-78
DDC

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 11	2. GOVT ACCESSION NO.: --
3. RECIPIENT'S CATALOG NUMBER: --	
4. TITLE (and Subtitle): A low-cost, general purpose data acquisition and control system for the PDP-11 minicomputer.	5. TYPE OF REPORT & PERIOD COVERED: Technical [redacted] rept.
	6. PERFORMING ORG. REPORT NUMBER: TR-11
7. AUTHOR(s): J. D. S. Danielson, Steven D. Brown, Carl J. Appellof [redacted] B. R. Kowalski	8. CONTRACT OR GRANT NUMBER(s): N00014-75-C-0536
9. PERFORMING ORGANIZATION NAME AND ADDRESS: University of Washington, Seattle, Washington 98195	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: NR 051-565
11. CONTROLLING OFFICE NAME AND ADDRESS: Department of Chemistry BG-10, University of Washington, Seattle, Washington 98195	12. REPORT DATE: Feb 10, 1978 (10 Feb 78) 13. NUMBER OF PAGES: 21 (23p.)
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): --	15. SECURITY CLASS (of this report): UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

--

18. SUPPLEMENTARY NOTES

Prepared for publication in: Chemical Instrumentation

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A general-purpose interface for the PDP-11 family of minicomputers is described. The interface, oriented towards laboratory data-acquisition and experimental control applications, can be built in a relatively short time with low materials costs. A general purpose, FORTRAN-compatible software package capable of driving the interface is also discussed.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

370 274

OFFICE OF NAVAL RESEARCH

Contract N00014-75-C-0536

Task No. NR 051-565

A Low-Cost, General-Purpose Data Acquisition and Control System For The PDP-11 Minicomputer

Prepared for Publication

in

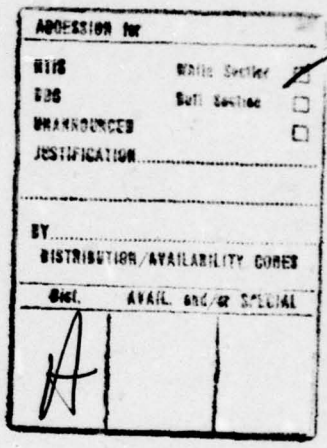
Chemical Instrumentation

by

J. D. S. Danielson
Steven Brown
Carl J. Appellof
and
B. R. Kowalski

Department of Chemistry BG-10
University of Washington
Seattle, Washington 98195

February 1978



A LOW-COST, GENERAL-PURPOSE DATA ACQUISITION AND CONTROL SYSTEM FOR THE PDP-11 MINICOMP

J.D.S. Danielson, Steven D. Brown, Carl J. Appellof and B. R. Kowalski

Department of Chemistry BG-10
University of Washington
Seattle, Washington 98195

ABSTRACT

A general-purpose interface for the PDP-11 family of minicomputers is described. The interface, oriented toward laboratory data-acquisition and experimental control applications, can be built in a relatively short time with low materials costs. A general purpose, FORTRAN-compatible software package capable of driving the interface is also discussed.

The past five years have seen the gradual emergence of several different approaches to laboratory computing. One approach, that of a dedicated computing system interfaced to a single instrument, is by far the most popular.[1-5] Both minicomputer-[1-3] and microprocessor-based[4,5] systems have been proposed or used for virtually every imaginable instrument, and each of these two competing schemes has its advantages. Microprocessor-based systems are frequently much cheaper than minicomputer-based systems, but the smaller (and cheaper) microprocessors can be troublesome to program, as many do not support higher-level languages. In either case, the interface between the computer and instrument usually is specific to that particular instrument, and generally is not easily changed when either the instrument is modified or the experiment using the instrument is concluded. Changes require an extensive knowledge of both digital and analog electronics as well as an ability to program in the particular assembly language peculiar to the computer involved. All of these considerations are involved in the decision to build an interface, particularly so when one wishes to maximize the cost-effectiveness of the particular set-up.

A second approach, that of using hierarchical computing systems,[6-8] appears to meet the programming problem discussed above, and to some extent minimizes the cost involved by allowing a more flexible use of a larger computer. Hierarchical systems are usually built around one or more microprocessors, each interfaced to a single instrument, with the microprocessors controlled by a portable minicomputer. One can use minicomputer-based cross-assemblers to program the microprocessor,[8] and one gains the computing power of the minicomputer without dedicating it to one instrument, thus spreading the minicomputer's cost over, hopefully, several applications. The interface between instrument and computer,

in this case a microprocessor, remains a single-purpose unit; however, modifications and/or changes in the instrument involved in the interface still require knowledge of electronics.

The original[9] approach to the problem of interfacing, that of using a general-purpose interface[10-12] had received considerably less attention until recently.[7,13] Part of the reason for the use of the other systems mentioned above is that the cost advantage theoretically achieved by a general-purpose interface in allowing interfacing to a variety of instruments is nullified by the considerable cost of some commercial interfaces. Additionally, many of these interfaces are not suited to rapid data acquisition nor to convenient programming in a higher-level language, so that the user must still program in assembly language and must now include the proper protocol for the general-purpose interface as well. Recent products, such as the IBM Device Coupler, meet these problems, but lack the ability to transfer data rapidly or to allow accurate timing in time-critical experiments. Few general-purpose interface designs[10] have appeared in the literature, no doubt because of the time involved in developing and building such a device. This paper describes the design of and software support for an interface suited to all PDP-11 minicomputers. It is sufficiently general to allow a variety of laboratory applications, while retaining a relatively simple design to allow rapid construction. The parts cost for the entire interface was under $3,500 and about 1500 man-hours were required for the entire project, including design and testing. In addition to the general purpose hardware to be described below, we have also developed a series of assembler-based subroutines for the higher-level languages supported by our system. This frees most users from the need to have either a detailed understanding of the interface electronics or to have the ability to program in PDP-11 assembly

language. These facets, plus the relative portability of the system, allow a user to gain the advantages of computerized experimental control and data acquisition in a minimum of time.

Design Considerations

Design of a computer interface must take the actual computer architecture into consideration. The PDP-11 is a 16-bit, parallel-logic computer using two's complement arithmetic. All communication between system components is done on a single high speed bus called the UNIBUS, which has separate lines for address, control and data signals. Each device is assigned a specific address on the UNIBUS; this allows flexible manipulation of all devices by the central processor, and makes interfaces to a PDP-11 particularly straightforward. Along with normal communication on the UNIBUS, interrupts are allowed, with a variety of priority levels being possible. Most PDP-11 processors distinguish levels 4-7 with 7 being the highest (excluding the non-processor request, which has the highest priority). Assignment of a particular priority level to a device places a relative importance on the function of that device, such as collecting data, timing events, and so forth.

Our interface consists of six devices on the UNIBUS. In order to explain the workings of our system, we will discuss each device in turn, omitting circuit diagrams, but detailing the function of the registers controlling each device. All devices have 18-bit addresses to allow compatibility with larger PDP-11 computers. All addresses and interrupt vectors are jumper selected to allow modifications if the need arises, or to tailor the interface to a particular system.

Data Acquisition

The heart of any computer interface is its analog input capacity. Our interface uses a Datel MDAS-16 Data Acquisition System which consists of a 16-channel analog multiplexer,

monolithic sample and hold amplifier, and a 6 μsec, 12-bit successive approximation analog-to-digital converter. The input impedence is 100 MΩ, and the input voltage may be converted either on the ±10, ±5, ±2.5V bipolar scale range, or on the +10 or +5 unipolar scale range. The maximum throughput rate is 57kHz at ± .025% of full-scale response error.

This unit allows us 16 single-ended channels of input. The pin-compatible MDAS-8D can be used in its place if differential inputs are desired. Our inputs have been divided into two groups of eight each, with the first set having switchable voltage sources controlled by potentiometers on the front panel in addition to an external input, allowing either external or internal signals to be selected. The second set handles only external signals. All external inputs are through BNC panel jacks.

The control and data registers for the ADC unit are shown in Figure 1. The ADC can be enabled to interrupt at priority level 7 upon an error, upon completing a conversion, or both. It may either be started by the program or by the underflow of clock #2 (see below). An extra data latch provided for the ADC allows ADC restart without a loss of data that would otherwise be caused by the new conversion. These features, as well as the extended sign on the data register, allow easy programming and very fast sampling rates.

Logic Levels, Logic Pulses and Digital Display

The second device contained in the interface is a collection of three different functions packed into two control registers to most effectively use available address decoding logic, and to minimize construction time, without significantly increasing programming difficulty. Figure 2 shows the control registers for this device.

Four logic levels are provided, as are four logic pulses. The logic levels provide a +3.5V output when a logic "1" is

Fig. 1
Fig. 2

loaded into the bit corresponding to that level; a logic "0" provides 0.0V output from the corresponding logic level. The logic levels, and all other outputs for this interface, have buffered outputs for isolation and for greater drive capability. The logic pulses, based on TTL 74123 (one-shot) chips provide a 1 μsec, + 3.5V pulse when the corresponding bit changes from logic "0" to logic "1." Output is through BNC panel jacks. The logic levels are useful for a variety of applications, including changing gains of instrumentation, while the logic pulses can be used to control stepper motors or to synchronize external instrumentation to the computer.

A six-digit decimal display occupies the lower 12 bits of each of the two control registers for this UNIBUS device, three digits residing in each of the registers. Coding for the output Hewlett-Packard HP 5082-7300 chips is in binary coded decimal (BCD), and thus 6 decimal digits require 24 binary bits to display. Conversion of binary to BCD is not performed by the hardware, but must be done by the program.

Relays and Switched Outlets

This interface is equipped to control a variety of instrumentation via the usual SPDT reed relay (Claire 922A12C1C) function as well as by solid-state switched 115V AC (Crydom D1210) outlets. Relays and outlets may be turned on by setting the appropriate bit in the control register; clearing the bit turns the relay (or outlet) off. Figure 3 details the relay and outlet control register.

Analog Output

Four digital-to-analog converters are provided to allow for a variety of uses. Their control registers are shown in Figure 4. The Burr-Brown DAC 80-CBI-V, 12 bit DAC was chosen

Fig. 3
Fig. 4

because of its good resolution, linearity and settling time, as well as its low cost. Each of the DAC's has been equipped with a front-panel range select switch, allowing bipolar operation in the ±10, ±5 and ±2.5V ranges and unipolar operation in the +10 and +5 V ranges. Outputs are BNC panel jacks.

Digital I/O, Sense Switches, Contact Closures and Schmitt Triggers

This interface includes the capability of rapid parallel digital I/O as well as a variety of external logic input controls in the logic input control registers. These are described in Figure 5. The four Schmitt triggers, based on LM 311H comparators, have variable threshholds, may be AC or DC coupled to the external instrumentation, and are equipped with a front-panel light to indicate to the operator when the trigger fires, for easy threshhold adjustment. A trigger firing sets the corresponding bit in the trigger register, and if the appropriate interrupt enable is set in the control and status register, causes an interrupt at priority level 5.

The eight illuminated sense switches (Unimax 01-282) also may be used to interrupt the program at level 5, or they can cause program branches by a software flag polling procedure. The eight contact closures set a specified bit when an external microswitch closes, again allowing either interrupts (if desired) at level 5 or flag polling. Also included in the logic device above is the digital parallel I/O port. The unit is set up in a handshake configuration to allow easy interprocessor communication. Data can also be strobed into the output word for use by external devices capable of decoding a 16-bit digital word, such as a monochromator or a parallel graphics terminal interface. In all of the above, if an

interrupt occurs, a software poll of the logic control and status register must also be performed to determine first which function caused the interrupt, then the control register for that function must be polled to determine second the exact channel requesting the interrupt. This software polling, while awkward from a programming standpoint, allows a simpler and cheaper design, which sacrifices only a little speed.

Real Time Clocks

The interface is equipped with three programmable clocks, shown in Figure 6. Clocks 0 and 2 are variable time base clocks for general purpose use, while clock 1 is a combination clock ^{14} for use in measurements of elapsed time. The clocks, based on the Mostek MK 5009 P chip, all have one-word time-base, countdown buffer registers to allow easy programming. Clock 2 is set up to automatically start the ADC upon its underflow provided the clock start bit (bit 2) is set in the ADC control status register. Basic periods available range from the oscillator time of 10^{-6} sec to as long as 100 sec, scaled by decades. A countdown buffer is also provided to allow any period between the basic ones provided. The clocks automatically decrement this buffer, and reload it upon underflow except when in the elapsed time mode. The 12-bit capacity allows buffer words of up to 4096_{10}. Additionally, the MK5009 P counter allows special periods based on hours, minutes and seconds, and so a great variety of periods are easily achieved. Clock 1 may also be used for elapsed time by setting bit 15. All clocks are capable of interrupting at priority level 6 if enabled.

In addition to the features described above, the interface is equipped with unswitched 115V AC outlets and a Digital Equipment Corp. DL-11 serial I/O port capable of operation at

Fig. 6

speeds ranging from 1200 to 9600 baud. This port is used for fast communications between our PDP-11/05 and the larger PDP-10 and CDC 6400 computers on campus, allowing even higher-level computing,[16] and is also used for outputting graphics to our Tektronix 4012 terminal.

The interface physically consists of three quad-height boards and two hex-height boards, held in a modified BA11-ES extension box built into a DEC 19-inch relay rack. The external electronics for the interface are held in three 19-inch panels, also mounted in the rack. Power for the interface is supplied by a +5, ±15, ±24V supply housed in the rack.

The rack also holds the remainder of our laboratory computing system, which includes the PDP-11/05 processor, with 24K core memory, a RX-01 dual-drive floppy disk, the VR-14 GT-40 display, and a Gould 5000 printer-plotter interface. The Gould printer, the Tektronix terminal and the lines to external computers are the only parts not directly portable with the rack. Figure 7 shows the architecture of our system.

System Software

To complement the flexible hardware of the interface, we have developed a series of assembly-language-based routines to allow interface operations from high level languages. Both BASIC and FORTRAN are supported in this manner. Most reports discuss this same feature for BASIC,[12] but we prefer FORTRAN because of its greater speed and flexibility. With FORTRAN, user supplied assembly language subroutines may be added easily, either to the library of general purpose assembly subroutines we have developed for FORTRAN or to our collection of separate, more specialized subroutines. RT-11 FORTRAN also has specialized routines designed to allow higher-language control of specific addresses and bits.[17,18] FORTRAN file

Fig. 7

management capabilities are also particularly suited to a system with extensive storage capabilities such as ours. We routinely use BASIC in checking experimental setups, or to test various interface functions, as the immediate mode of BASIC is particularly useful in these cases. Table I lists FORTRAN-callable routines available for the interface. Our BASIC-callable routines are called in a manner essentially identical to that used for FORTRAN-callable assembly routines.

TABLE I

MACRO-11* Based Routines for FORTRAN Interface Control

Call Sequence	Action
Call ADC (C,IV,N,R,B)	Opens ADC channel C to sample N points at a sampling period of B x 10^R μsec, storing them segmentally as IV(±) integer variables.
Call DAC (C,IV)	Outputs analog voltage corresponding to the value of variable IV through DAC channel C.
Call DIS (IV)	Displays variables IV value in digital display.
Call PWR (X,....,S)	Turns on channels X,... (S=1), or turns off channels X,... (S=0).
Call RLY (X,...,S)	As above, except S=1 closes relays, while S=0 opens relays.
Call PULSE (X)	Fires 1 pulse on channel X of logic pulses.
Call CLOCK (C,R,B)	Enables interrupt by clock on channel C at period of B x 10^R μsec.
Call TRIG (X)	Enables interrupt on firing of Schmitt trigger X.
Call WAIT (C)	Waits for interrupt for: clock, C=1; Schmitt trigger, C=2; closure, C=3.
Call CLEAR	Clears interrupt enables and interrupt vectors.

Call: WORDOUT ("VAR)	Passes digital word to digital I/O port (octal)
WORDIN ("VAR)	Accepts digital word from digital I/O port (octal)
Call LEVL (X,...,S)	Turns "on" logic levels X,...(S=1) or Turns "off" logic levels X,...(S=0)
A = SENS(N)	Function returns value 1 if sense switch N is set, 0 if not set.

*MACRO-11 is a PDP-11 Assembly language

Conclusion

In the six months that the interface has been in operation, we have experienced no problems. It has performed as expected in all applications tried to this date, and has required no special debugging or maintenance since its installation. A variety of users are either running experiments, or are developing experiments using the interface.

ACKNOWLEDGMENTS

The authors thank M. Champion for assistance in the construction and debugging of the interface. This work was supported in part by the Office of Naval Research.

REFERENCES

1. E. Lundberg and G. Johanson, Anal. Chem., 48, 1922 (1976).
2. T. A. Nieman and C. G. Enke, Anal. Chem., 48, 619 (1976).
3. B. W. Renoe, K. R. O'Keefe and H. V. Malmstadt, Anal. Chem. 48, 661 (1976).
4. S. P. Avery and H. V. Malmstadt, Anal. Chem., 48, 1309 (1976).
5. R. W. Hendler, D. Songco and T. R. Clem, Anal. Chem., 49, 1909 (1977).
6. J. A. Perry, M. F. Bryant and H. V. Malmstadt, Anal. Chem., 49, 1702 (1977).
7. G. E. Mieling, R. W. Taylor, L. G. Hargis and H. L. Pardue, Anal. Chem., 48, 1686 (1976).
8. P. D. Gaarenstroom, J. P. English, and S. P. Perone, ONR Tech Report 13, 1977.
9. S. P. Perone, Anal. Chem., 43, 1288 (1971), and references therein.
10. L. Ramaley and G. S. Wilson, Anal. Chem., 42, 606 (1970).
11. J. E. Davis and E. D. Schmidlin, Chem. Inst., 4, 169 (1973).

12. G. W. Bushnell, T. K. Davies, A. D. Kirk and S.K.P. Wong, Anal. Chem., 48, 587 (1976).
13. A. Guido, H. Cole and L. B. Kreighbaum, Proc. IEEE, 63, 1509 (1975).
14. B. K. Hahn and C. G. Enke, Anal. Chem., 45, 651A (1973).
15. PDP-11 Processor Handbook, Digital Equipment Corporation, 1973.
16. R. E. Dessy, Anal. Chem., 49, 1100A (1977).
17. PDP-11 RT-11 System Reference Manual, Digital Equipment Corp., 1975.
18. PDP-11 FORTRAN Reference Manual, Digital Equipment Corp., 1975.

FIGURE CAPTIONS

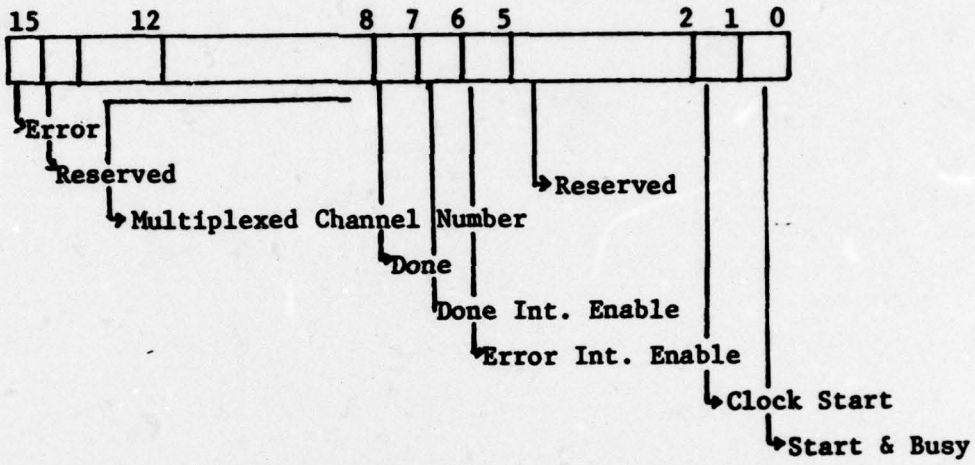
Captions are already included on the typewriter-produced figures. Figure 7 has been photoreduced, and should bear the following caption:

Figure 7. Architecture of PDP-11 with General Purpose Interface.

Figure 1

ADC Control and Data Registers

ADC Control Register (address 764020)



Bits	Function
0	Start ADC & indicate ADC not finished
1	Start ADC with clock 2 underflow
2-4	Reserved
5	ADC error causes interrupt through address 130
6	ADC done causes interrupt through address 130
7	Set when ADC done cleared upon reading data out
8-11	Loaded with 0-18 to access any of 16_{10} channels
12-14	Reserved
15	ADC error

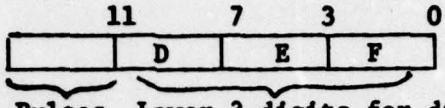
ADC Data Register (address 764022)

Bits	Function
0-10	Converted voltage magnitude
11-15	Converted voltage sign

Figure 2

Logic Levels, Logic Pulses and Digital Display Control Registers

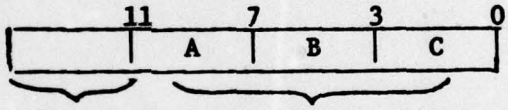
Pulse and Low Order Decimal Digits Control Register (address 764032)



Pulses Lower 3 digits for display (BCD code)

Bit(s)	Function
0-3	Digit F of ABCDEF
4-7	Digit E of ABCDEF
8-11	Digit D of ABCDEF
12	Logic Pulse 0
13	Logic Pulse 1
14	Logic Pulse 2
15	Logic Pulse 3

Level and High Order Decimal Digits Control Register (address 764034)



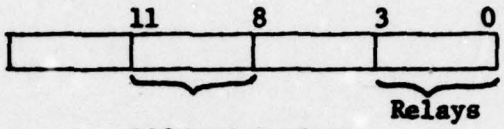
Levels Higher 3 digits for display (BCD code)

Bit(s)	Function
0-3	Digit C of ABCDEF
4-7	Digit B of ABCDEF
8-11	Digit A of ABCDEF
12	Logic Level 0
13	Logic Level 1
14	Logic Level 2
15	Logic Level 3

Figure 3

Relay and Switched Outlet Control Register

(Address 764030)

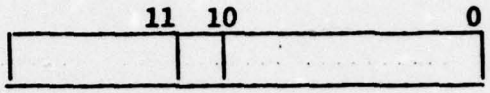


Bit	
0	Relay 0
1	Relay 1
2	Relay 2
3	Relay 3
8	Outlet 0
9	Outlet 1
10	Outlet 2
11	Outlet 3

Figure 4

Digital-Analog-Converter Data Registers

(Addresses 764060-764066)

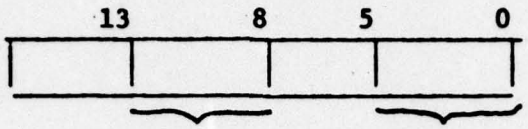


Bit(s)	Function
0-10	Data to be converted
11	Bipolar range - sign of number Unipolar range - most significant bit of data

Figure 5

Digital Logic I/O Registers

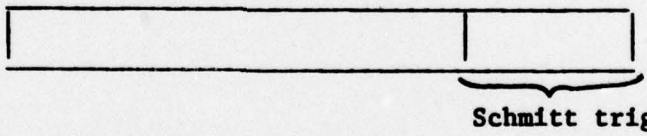
Control and Status Register (Address 764040)



Bit	Function
0	Schmitt trigger interrupt enable
1	Contact closure interrupt enable
2	Sense switch interrupt enable
3	Inword flag interrupt enable
4	Outword requested interrupt enable
8	Schmitt trigger event flag
9	contact closure event flag
10	Sense switch event flag
11	Input word ready flag
12	Output word requested flag

Interrupt vector is 110; priority is 5.

Schmitt Trigger Register (Address 764042)

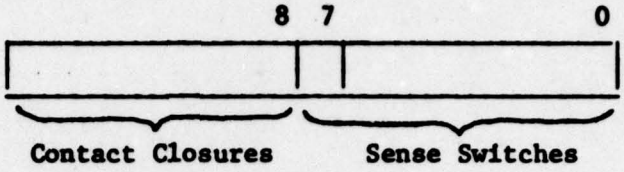


Bit	Function
0	Schmitt trigger 0
1	Schmitt trigger 1
2	Schmitt trigger 2
3	Schmitt trigger 3

Bits are set to indicate which trigger fired.

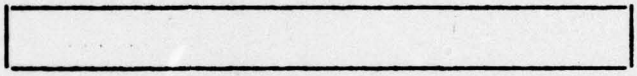
Figure 5, Continued

Sense Switch and Contact Closure Register (Address 764044)



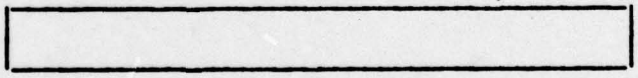
Bit	Function
0 (8)	Sense switch 0 (contact closure 0)
1 (9)	Sense switch 1 (contact closure 1)
2 (10)	Sense switch 2 (contact closure 2)
3 (11)	Sense switch 3 (contact closure 3)
4 (12)	Sense switch 4 (contact closure 4)
5 (13)	Sense switch 5 (contact closure 5)
6 (14)	Sense switch 6 (contact closure 6)
7 (15)	Sense switch 7 (contact closure 7)

Input 16-Bit Word Register (Address 764046)



Bits	Function
0-15	Data

Output 16-Bit Word Register (Address 764050)

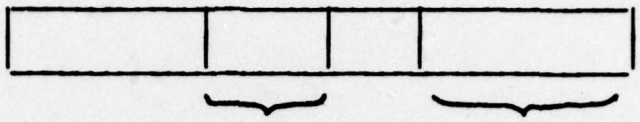


Bits	Function
0-15	Data

Figure 6

Programmable Clock Registers

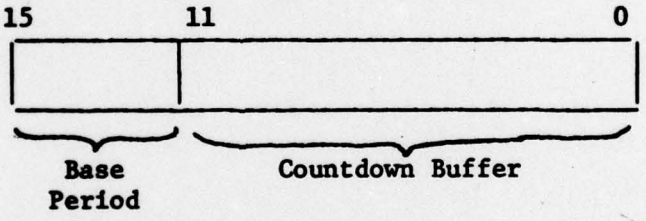
Control and Status Register (Address 764100)



Bit	Function
0	Start clock 0
1	Enable interrupt for clock 0 on underflow
2	Start clock 1
3	Enable interrupt for clock 1 on underflow
4	Start clock 2
5	Enable interrupt for clock 2 on underflow
8	Clock 0 underflow flag
9	Clock 1 underflow flag
10	Clock 2 underflow flag

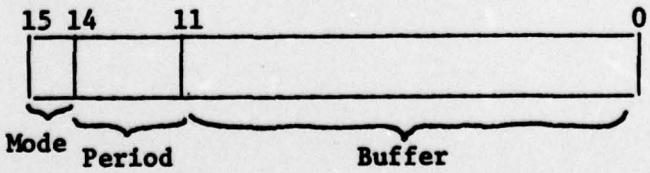
Interrupt vector is 102; priority is 6.

Clock Base Period and Countdown Buffer Registers (Addresses 764102 and 764106)
(For variable time base clock)

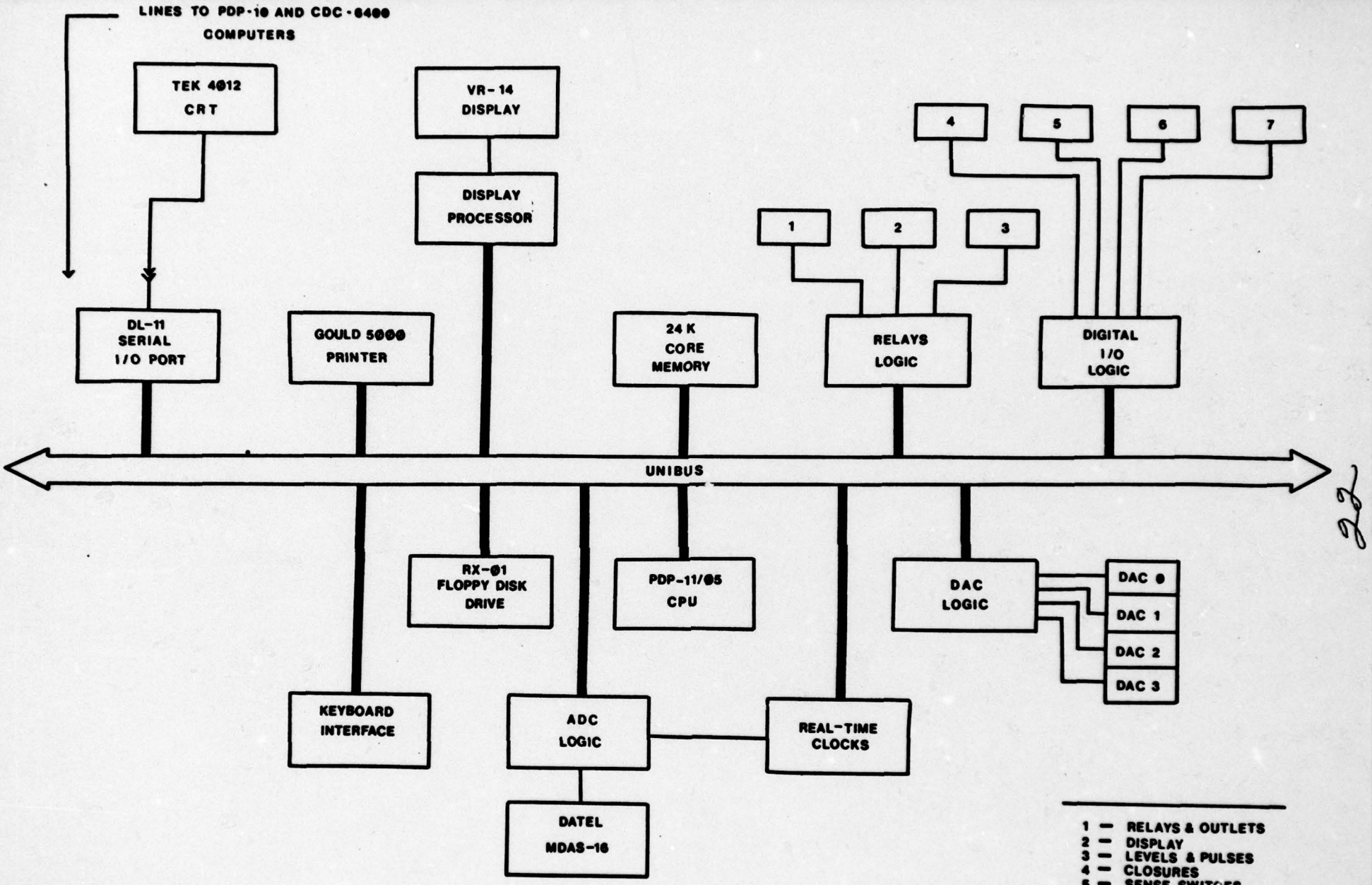


Bits	Function
0-11	Write only countdown buffer
12-14	Clock period decade multiplier
15	Hours, minutes and seconds selector

Combination Clock Buffer (Address 764104)



Bits	Function
0-11	Buffer for clock (read-write)
12-14	Clock period multiplier
15	Mode bit (elapsed time=1)



Chemometrics Laboratory Computing System